



# درس طراحی زبان‌های برنامه‌سازی

دکتر محمد ایزدی

تمرین سوم

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۱۴۰۰-۱۳۹۹

---

مهلت ارسال:

۱۶ اردیبهشت ۱۴۰۰

ساعت ۲۳:۵۹



### به موارد زیر توجه کنید:

- \* برنامه‌های خود را به زبان Racket بنویسید.
- \* مهلت ارسال تمرین ساعت ۲۳:۵۹ روز ۱۶ اردیبهشت ۱۴۰۰ است.
- \* جواب خود را در قالب pdf بنویسید. در صورت نیاز می‌توانید کدهای خود را در کنار pdf بگذارید.
- \* در نهایت تمام فایل‌های خود را در یک فایل زیپ قرار داده و با نام *HW3\_StudentID* در سامانه کوئرا آپلود کنید.
- \* هرگونه سوالی راجع به این تمرین را در زیر پست مربوطه در کوئرای درس مطرح کنید.
- \* در مجموع تمامی تمرین ۷ روز مهلت تاخیر مجاز دارید و پس از تمام شدن این تاخیرهای مجاز به ازای هر روز ۱۰ درصد از کل نمره تمرین شما کم می‌شود.
- \* لطفا تمرین‌ها را از یکدیگر کپی نکنید. در صورت وقوع چنین مواردی مطابق با سیاست درس رفتار می‌شود.
- \* بخش سوال‌های پیشنهادی نیازی به تحویل ندارد و صرفاً جهت تمرین بیشتر گذاشته شده است.



## سوال ۱

Abstract Syntax و Concrete Syntax را تعریف کرده و با ارائه‌ی یک مثال تفاوت آن‌ها را بیان کنید.

## سوال ۲

نوع داده‌ی صف<sup>۱</sup> را در نظر گرفته و برای توابع زیر یک توصیف<sup>۲</sup> نوشته و سپس این توابع را پیاده‌سازی کنید.

- empty-queue — Return a new empty queue.
- enqueue — Add an item to the queue.
- dequeue — Remove an item from the queue.
- peek — Gets the element at the front of the queue without removing it.
- isfull? — Checks if the queue is full.
- isempty? — Checks if the queue is empty.

نحوه‌ی پیاده‌سازی و جزئیات آن (همانند ورودی‌های مورد نیاز توابع یا چگونه در نظر گرفتن طول برای این صف) بر عهده‌ی خودتان است.

## سوال ۳

نوع داده‌ی لیست پیوندی دوطرفه<sup>۳</sup> را در نظر گرفته و توابع زیر را پیاده‌سازی کنید.

- empty-doubly-linked-list — Return a new empty doubly linked list.
- insert-first — Add an element at the beginning of the list.

<sup>۱</sup> queue  
<sup>۲</sup> specification

<sup>۳</sup> [https://www.tutorialspoint.com/data\\_structures\\_algorithms/doubly\\_linked\\_list\\_algorithm.htm](https://www.tutorialspoint.com/data_structures_algorithms/doubly_linked_list_algorithm.htm)



- delete-first — Delete an element at the beginning of the list.
- insert-after — Add an element after an item of the list.
- insert-before — Add an element before an item of the list.
- delete — Delete an element from the list using the key.
- isfull? — Checks if the list is full.
- isempty? — Checks if the list is empty.

نحوه‌ی پیاده‌سازی و جزئیات آن بر عهده‌ی خودتان است. دقت کنید مهم‌ترین ویژگی این داده‌ساختار حرکت روی نودها به چپ و راست است. در واقع هرگاه بخواهیم از یک نود به چپ یا راست برویم، در صورت وجود، آن نود برای ما مشخص است.

## سوال ۴

نوع داده‌ی محیط <sup>۴</sup> را که در کلاس توضیح داده شده است به گونه‌ای پیاده‌سازی کنید که دارای ویژگی‌های زیر باشد:

- هیچگاه ارور ندهد و در صورت نبودن داده در آن، مقدار 'not-set-variable' برگردانده شود.
- تابع has-binding? داشته باشد. این تابع با گرفتن یک متغیر و یک محیط، در صورت موجود بودن آن متغیر در آن محیط، #t و در غیر این صورت #f برگرداند.
- تابع union را پشتیبانی کند. این تابع با گرفتن دو محیط، دو محیط را اجتماع می‌گیرد و نتیجه را برمیگرداند. در صورت وجود یک متغیر در هر دو محیط، مقدار متغیر در محیط دوم باید برگردانده شود.

## سوال ۵

در این سوال از زبان eopl استفاده خواهیم کرد. با استفاده از define-datatype محیط تعریف شده در سوال ۴ را تعریف کنید.

<sup>۴</sup>environment



## سوال‌های پیشنهادی

سوال‌های زیر از کتاب *essentials of programming languages* توصیه می‌شود:  
2.4, 2.12, 2.19, 2.20